1/PRTS

10/528787
JC14 Rec'd PCT/PTO 23 MAR 2005

MULTIPLE MATCHING CONTROL METHOD

The present invention refers to the domain of pairing between a security module and a host module, with the particular aim of securing communications between the two modules.

Pairing is a known mechanism that consists in dividing a unique secret between two devices thus rendering the communication between these two devices inaccessible to all other devices.

This pairing is described in application EP1078524 and allows the connection of a security module to a receiver thanks to the presence of a unique encryption key known only by these two elements.

In an environment that allows the connection of a security module to several host apparatuses such a pairing is not possible, as it is too restrictive.

The document WO02/052515 describes a solution that puts into practice the pairing control by means of a management centre. The security module can be paired to any apparatus as long as the management centre gives authorisation. This solution supposes the existence of a channel that allows the management centre to send one or more messages to the security module.

Therefore, the aim of this invention is to pair a security module with one or more host apparatuses in an environment in which the call to a management centre is not possible at the time of the pairing, that is to say, there is no channel between the management centre and the security module.

This aim is achieved thanks to a pairing control method between a first device such as a removable security module and a second device such as a host apparatus, this pairing consisting in securing data exchanges with the aid of a unique pairing key, this method consisting in:

- verifying the pairing between the two devices and using the unique pairing key if the pairing has been already carried out , if not,

- searching for a free location among the locations reserved for the pairing data in the first device and in this case,

- initiating a pairing procedure by transmitting a cryptogram contained in the second device and that contains an identifier belonging to this device, this cryptogram being encrypted by a secret key common to all the first devices,

- decrypting this cryptogram using the first device and extracting from this cryptogram the identifier of the second device,

- generating a pairing key based on this identifier,

- storing in the first device the pairing data with the second device.

This method contains two important characteristics. The first is the possibility of storing several pairing data in the security module (first device). The maximum number will be voluntarily limited in order to prevent the same module pairing with an unlimited number of host apparatuses.

The second characteristic is the way in which the pairing key is created. Initially, one particular security module is not destined to pair with a particular host apparatus. This is why, according to a first variant, a unique identifier is encrypted in the host apparatus (second device) with a key that is contained in each security module. This identifier can be the serial number of the host apparatus, an encryption key or a number randomly generated during the personalization of each host apparatus or it can be a mixture of these elements.

According to an embodiment, the cryptogram contains a secret key that can be of the symmetrical or asymmetrical type. Once decrypted by the security module, the latter generates a random key that will be the pairing key and encrypts it with the secret key then sends it to the host apparatus. The unique serial number of the host apparatus will preferably be contained in the first messages exchanged between the two elements in order to obtain pairing verification.

In a second embodiment, the pairing key is already included in the cryptogram transmitted by the second device. In this case, the pairing key is a unique key,

belonging to the host apparatus and does not depend in any way on the security module.

The invention also refers to a way in which the cryptogram is contained in the security module. The latter is that which will transmit the cryptogram to the host apparatus for the generation of the pairing key. It is to be considered that the common decryption key, in this case stored in the host apparatus, is stored in a security element, such as a secured memory.

If a new pairing is carried out, the pairing data will be registered and will occupy one of the locations envisaged for the different pairing that a security module is able to accept. The pairing data is for example the host apparatus serial number together with the pairing key.

Due to the fact that the number of locations is limited, it is probable that the security module will be connected to a new host apparatus while all the locations are in use. To determine the location to be replaced, there are several mechanisms, namely:

- an activity counter associated to each location. At each pairing negotiation between the security module and the host apparatus this counter is increased. In this way, the smallest counter determines the location least used. Said location is that which will be replaced by the new pairing. Pairing negotiation is generally understood to mean the powering on the host module and the request for information by the security module.

- a pairing chronology counter associated to each location. At each pairing negotiation, the corresponding counter takes the value of the greatest of all the counters plus one, except if this counter is already the greatest, in which case it is not modified. Thus, the counter having the lowest value indicates the location of the oldest pairing. This is the location that will be replaced by the new pairing.

In one embodiment, with any new pairing or any pairing changes (this happens when no free locations are available) a secret code (PIN code) is introduced. On the first insertion of the security module in the host apparatus the security module initiates a sequence in the host apparatus that, according to its display means, requests the user to introduce this secret code. When the user introduces the correct code, which

is then transferred towards the security module, it is the only valid case for which the latter will accept this new pairing.

According to the chosen variants, it is possible that this secret code will be required for each new pairing without relation to the occupancy of the memory locations. In another variant, it is possible to force the secret code to intervene in the case of replacing a location that is already occupied.

Several variants are envisaged to determine the validity of this secret code. In a first simplified variant, the secret code is constant for a security module and is distributed with said module.

In a second variant, the user calls or connects to a management centre that transmits to the user the unique number of the security module and of the host apparatus. This centre calculates a secret code according to an algorithm taking into account the two variables that are the two unique numbers. This algorithm is also contained in the security module in order to verify the conformity of the secret code. The call to the management centre can be made prior to pairing so that the necessary code will be available when the module connects with the host apparatus.

According to a third variant, the algorithm used for the calculation of the code is based on the unique number of the security module and of an incremental index. This code is then combined with the unique number of the host apparatus in order to obtain the secret code that is then transmitted to the user to authorize its new pairing.

The code can be determined according to the formula: CS = G(K, (FN(UA))) = G(K, F((FN-1(UA)))), in which CS is the secret code, UA the unique number of the security module, N the incremental index, K the unique number of the host apparatus, F an encryption function and G a function which makes K intervene in the calculation of the CS.

In this way, the secret code inevitably changes for each pairing. Either the result of the function FN-1(UA), or the value of the index N is stored in the module memory to be used as the starting point for the next pairing. In order for the centre to be able to calculate the correct secret code, it is necessary for the centre to be synchronised

with the security module. For this, the user, during the request, can for example, transmit to the centre the value of the index N or the result of the function FN-1(UA) previously transmitted by the security module. Of course, the user must also transmit the unique number of the security module and of the host apparatus to the management centre.

However, if the value of the index N in the security module is not accessible to the management centre, said centre can transmit a secret code that does not necessarily correspond to the last index of the security module. Due to this eventual difference between the index stored in the security module and the index stored in the management centre, a secret code correctly calculated in the management centre can be rejected by the security module.

In this case, it is possible to resynchronise the security module. If for example, the management centre has provided a secret code originating from the number of the user's host apparatus and from the cryptogram of incremental index 12, that is to say that which is in the management centre, and if the cryptogram stored in the security module is of index 8, then the module will calculate the secret codes corresponding to indexes 8, 9, 10, 11, 12 to notice that the cryptogram originating from the manually introduced code corresponds to a valid cryptogram of a higher index. This noticing indicates that the management centre has previously sent four secrets codes that the user of the security module has finally not used.

It is certain than the index difference between the current index (8 in our example) and the management centre index (12 in our example) will be limited to an acceptable number. It is not a question of searching through thousands possibilities in the hope of finding the correct secret code.

It is to be highlighted that this third variant includes the possibility of not allowing the intervention of the unique number of the host apparatus in the calculation of the secret code, by defining CS = (FN(UA)) which corresponds to the case in which the previously mentioned function G is defined using G(x, y) = y. This variant is interesting if one wishes to separate the secret code from the host apparatus number. In fact, if it is easy to find out the number of the security module, by definition an easily transportable module, it is more difficult to find out the unique

number of the host apparatus, in particular when the secret code has to be obtained before connecting the two elements.

The invention will be better understood thanks to the following detailed description that refers to the unique figure that is given as a non-limitative example and that shows the two main elements and the data that they contain.

The security module MS includes a secured database DB in which, amongst others, the pairing data is to be found. This reference data PDT1 to PDTn occupies the memory locations from 1 to n. Note that the number n of locations envisaged in the module MS can be equal to 1.

This base DB also contains the key k common to all the security modules MS and allows the decryption of the cryptogram CY as well as the index N of the number of pairings previously carried out.

Initially, the host module MH contains this index in a memory M that can either be of the secured type or freely accessible. However, it is preferable that this memory is protected and difficult to access in order to avoid one host apparatus being confused with another.

This cryptogram CY is encrypted by the key k and contains, in one embodiment, the serial number SN and a mark PT which value is known by the security module. This mark PT allows the security module ensuring the validity of the cryptogram. This mark PT is common to all the cryptograms. According to another variant, it can be unique to the host apparatus. The cryptogram CY can also contain the pairing key MHKey of the host apparatus that will later be used to secure the data transmission between the module MS and the host apparatus. For example, once this key is known by the two modules, a session key KS can be negotiated and used to encrypt the communication. Of course, in such a case, the key MHKey must also be stored in memory M of the host apparatus and this memory must therefore be secured.

In the database DB of the security module MS, the data PDT1 to PDTn includes an activity or chronology counter such as described above. It is to be remembered that these counters allow the determination of the location to be replaced in the case that all locations are in use. In the cases where activity counters are used, the three

locations can be used as an example, such as the locations PDT1 to PDT3 respectively occupied by the pairings carried out by host modules MHA, MHB and MHC. At each pairing negotiation between the security module MS and the module MHC for example, the counter CPT3 will be increased.

In the embodiments in which a session key KS generated from the pairing key KA is used, it should be noted that this pairing can evolve dynamically, that is to say, that the session key KS is necessary changed after a certain usage period; on the basis of the elements transmitted during the pairing between these two entities (pairing key, host module key MHKey), a new session key is generated in this way. The number of session keys already generated can then be counted and this number can be considered as an activity counter.

When a new pairing request is required to the security module, it will determine the lowest activity counter and clear this location. Of course, the security module also contains all the necessary information to calculate and verify the secret codes.